Contract Address

- Any contract has its own **unique address** that is **generated at deployment.**
- The contract address is generated **based on the address of the account that deploys the contract and the no. of transactions** of that account (nonce). It can't be calculated in advance.
- Address is a variable type and has the following members:
  - balance
  - If the address is declared **payable** it has two additional members:
    - transfer(): should be used in most cases as it's the safest way to send ether
    - send(): is like a low-level transfer(). If execution fails the contract will not stop and send() returns false;
  - call(), delegatecall(), staticcall()

Example: 0xf8e81D47203A594245E36C48e151709F0C19fBe8

Payable functions and contract balance

- A smart contract can receive ETH and can have an ETH balance **only if there's at least one payable function**.
- A contract receives ETH in multiple ways:
    a. Just by sending ETH to the contract address from another account.
    The contract must have at least one of the functions below:

    - receive() external payable - for empty calldata (and any value)
    - fallback() external payable - when no other function matches (not even the receive function).

    b. By calling a payable function and sending ETH with that transaction.

- The ETH balance of the contract is in possession of anyone who can call the transfer() built-in function